

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-16468

(43)公開日 平成8年(1996)1月19日

(51)Int.Cl.<sup>6</sup>

G 0 6 F 12/08

識別記号

庁内整理番号

D 7623-5B

F I

技術表示箇所

審査請求 未請求 請求項の数8 OL (全 21 頁)

(21)出願番号 特願平6-142790

(22)出願日 平成6年(1994)6月24日

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 右田 学

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

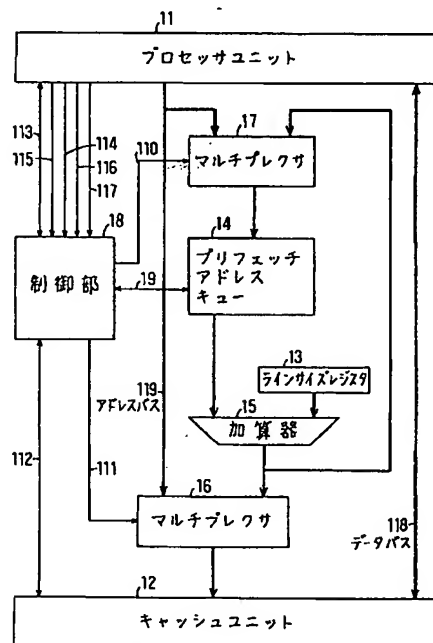
(74)代理人 弁理士 前田 弘 (外2名)

(54)【発明の名称】 データプリフェッチ制御回路

(57)【要約】

【目的】 データプリフェッチ制御回路において、1回のプリフェッチ要求で多数のラインデータのプリフェッチを可能とすると共に、多変数データ列のプリフェッチ動作を可能とする。

【構成】 プロセッサユニット11から出力されるプリフェッチアドレスはマルチプレクサ17を経てプリフェッチアドレスキュー14に格納される。ラインフィル動作の終了後、上記格納されたプリフェッチアドレスキューは、加算器15でラインサイズレジスタ13に記憶されたラインサイズ値と加算されて、更新されたプリフェッチアドレスが得られる。この更新値はマルチプレクサ17を経て再度プリフェッチアドレスキュー14に格納された後、キャッシュユニット12に出力されて、データのプリフェッチ動作が自動で行われる。以上の動作が繰返し行われる。



## 【特許請求の範囲】

【請求項1】 プロセッサユニットが使用するデータをキャッシュユニットにプリフェッチするデータプリフェッチ制御回路であって、プリフェッチアドレスを格納するプリフェッチアドレスキューと、

上記キャッシュユニットのラインサイズ値を格納するラインサイズ値格納手段と、

上記プリフェッチアドレスキューに格納されたプリフェッチアドレス及び上記ラインサイズ値格納手段に格納されたラインサイズ値に基いて、プリフェッチアドレスを

更新する演算手段と、  
上記プロセッサユニットから出力されるプリフェッチアドレス及び上記演算手段により更新されたプリフェッチアドレスのうち何れか一方を選択する第1及び第2の選択手段とを備え、

上記第1の選択手段は、選択した値を上記プリフェッチアドレスキューに出力し、上記第2の選択手段は、選択した値を上記キャッシュユニットに出力し、

更に、上記プロセッサユニットからプリフェッチアドレスが出力されるときこのプリフェッチアドレスを選択し、その後は、更新されたプリフェッチアドレスを選択するように、上記第1及び第2の各選択手段を制御する制御手段とを備えたことを特徴とするデータプリフェッチ制御回路。

【請求項2】 ラインサイズ値格納手段は、予め、所定の1種類のラインサイズ値が記憶されたレジスタにより構成されることを特徴とする請求項1記載のデータプリフェッチ制御回路。

【請求項3】 ラインサイズ値格納手段は、ラインサイズ値が外部から任意に与えられて、ラインサイズ値が可変であることを特徴とする請求項1記載のデータプリフェッチ制御回路。

【請求項4】 ラインサイズ値格納手段は、プリフェッチアドレスの増分値を格納する増分値格納キューと、

プロセッサユニットから出力されるオフセットデータ及び上記増分値格納キューに格納したプリフェッチアドレスの増分値のうち何れか一方を選択し、その選択した値を上記増分値格納キューに出力する第3の選択手段と、  
プロセッサユニットからオフセットデータが出力されるときそのオフセットデータを選択し、その後は、増分値格納キューに格納したプリフェッチアドレスの増分値を選択するよう、上記第3の選択手段を制御する第2の制御手段とから成ることを特徴とする請求項3記載のデータプリフェッチ制御回路。

【請求項5】 プリフェッチアドレスキューは、FIFOメモリであることを特徴とする請求項1記載のデータプリフェッチ制御回路。

【請求項6】 プロセッサユニットは、プリフェッチ中

断信号を出力し、

制御手段は、上記プロセッサユニットからのプリフェッチ中断信号を受けて、

プリフェッチアドレスキューに格納されているプリフェッチアドレスを消去することを特徴とする請求項1、請求項2及び請求項4記載のデータプリフェッチ制御回路。

【請求項7】 プリフェッチ回数をカウントし、プリフェッチ回数が設定回数に達したとき、プリフェッチ中断信号を出力する中断信号出力手段を備え、

制御手段は、上記中断信号出力手段から出力されるプリフェッチ中断信号を受けて、プリフェッチアドレスキューに格納されているプリフェッチアドレスを消去することを特徴とする請求項1、請求項2及び請求項4記載のデータプリフェッチ制御回路。

【請求項8】 中断信号出力手段は、

プリフェッチの実行残数を格納する実行残数格納手段と、

上記実行残数格納手段から出力される値をゼロ比較する比較手段と、

上記実行残数格納手段に格納されたプリフェッチの実行残数から1を減算する減算手段と、

プロセッサユニットから出力されるプリフェッチ回数及び上記減算手段により減算されたプリフェッチの実行残数のうち何れか一方を選択し、その選択した値を上記実行残数格納手段に出力する第4の選択手段と、

上記プロセッサユニットからプリフェッチ回数が出力されるときそのプリフェッチ回数を選択し、その後は、減算手段により減算されたプリフェッチの実行残数を選択するよう、上記第4の選択手段を制御する第3の制御手段とから成ることを特徴とする請求項7記載のデータプリフェッチ制御回路。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、データプリフェッチ制御回路の改良に関し、詳しくは、多変数データ列のプリフェッチを実現すると共に、プロセッサから発行されるデータプリフェッチ要求の回数を削減して、データのプリフェッチの効率化を図るようにしたものに関する。

【0002】

【従来の技術】近年、プロセッサとメモリデバイスとの性能格差を埋める目的で、プロセッサと主メモリとの間に、キャッシュメモリと呼ばれる高速なバッファメモリを設置して、プロセッサによる主メモリの実効的なアクセス時間を高速化するようにした計算機システムが広く普及している。

【0003】ところで、上記キャッシュメモリを備えた計算機システムでは、プロセッサからのロード要求があっても、キャッシュメモリ上にプロセッサの要求するデータが無い場合（以下、「キャッシュミス」という）に

は、キャッシュメモリの内容を更新するように、プロセッサの要求するデータを含んだ一定量のデータを主メモリから読み出す動作（以下、「ラインフィル動作」という）が行なわれる。

【0004】しかしながら、上記従来の計算機システムでは、ラインフィル動作中は、キャッシュメモリと主メモリとの間の整合性を保つために、プロセッサの処理は停止状態になる（以下、この状態を「ストール状態」という）。このため、ストール状態の継続時間の分、システム全体の性能が低下する欠点があった。特に、ライン

フィル動作時に主メモリからキャッシュメモリにデータ転送する場合のデータ転送サイズを増やして、キャッシュメモリのヒット率を向上させた場合には、ラインフィル動作に費やす時間が増加して、ストール状態が長く継続する。

【0005】そこで、従来、例えば特開昭62-192831号公報に開示されるものでは、ラインフィル動作時でのキャッシュメモリへのデータ転送サイズを比較的小さく設定して、プロセッサのストール時間を短縮すると共に、ラインフィル動作の終了後は、プロセッサの処理を継続しながら、そのプロセッサで次に利用する可能性の高いデータを主メモリからキャッシュメモリに予め読み出しておくプリフェッチ制御を行って、計算機システムの性能の向上を図っている。

【0006】以下、図面を参照しながら、上記従来のデータプリフェッチ制御回路の一例について説明する。

【0007】図13は従来のデータプリフェッチ制御回路の概略図を示す。

【0008】同図において、101はプロセッサユニット、102はキャッシュユニット、103は上記プロセッサユニット101から出力されるアドレスデータをラッチするアドレスレジスタ、104はキャッシュのラインサイズ（データ転送サイズ） $\alpha$ を格納するラインサイズレジスタ、105は上記アドレスレジスタ103及びラインサイズレジスタ104を用いて、プリフェッチアドレスの計算を行なう加算器である。

【0009】また、106は上記アドレスレジスタ103の出力と加算器105の出力との何れか一方を選択するマルチプレクサ、107はプリフェッチ動作を制御する制御部、108はプロセッサユニット101から出力されるプリフェッチ要求信号であって、この要求信号により、当初はラインフィル動作を行わせ、その後にプリフェッチ制御を行わせる。

【0010】更に、109は上記マルチプレクサ106への選択信号、1010はプロセッサユニット101と制御部107との間でハンドシェークされるアクセス制御信号、1011はキャッシュユニット102と制御部107との間でハンドシェークされるアクセス制御信号である。

【0011】上記データプリフェッチ制御回路につい

て、以下、その動作を説明する。

【0012】先ず、プロセッサユニット101がプリフェッチ要求信号108を通じて制御部107にプリフェッチ要求を発行すると、制御部107は、マルチプレクサ106がアドレスレジスタ103の出力（アドレス $n$ ）を選択するように選択信号109を設定すると共に、キャッシュユニット102に対してアクセス制御信号1011を用いてデータフェッチ要求を行う。従って、キャッシュユニット102は、上記データフェッチ要求に応じて所定のデータを主メモリから読出し、ラインフィル動作が行われる。その後、キャッシュユニット102は、所定のデータの用意ができた時点で、データバス1012にデータを出力すると共に、アクセス制御信号1011を用いて制御部107に動作完了信号を通知する。この通知を受けた制御部107は、アクセス信号1010を用いてプロセッサユニット101に対してデータフェッチ動作完了を通知する。プロセッサユニット101は、データバス1012のデータをフェッチして処理を再開し、この時点で初めて次の処理を行う。

【0013】続いて、制御部107は、マルチプレクサ106が加算器105の出力を選択するように選択信号109を設定する。一方、加算器105では、アドレスレジスタ103の内容とラインサイズレジスタ104の内容との加算を行い、その加算結果であるプリフェッチアドレス（ $n + \alpha$ ）をキャッシュユニット102に対して通知する。この要求により、キャッシュユニット102は、プリフェッチアドレス（ $n + \alpha$ ）を含むデータラインを主メモリから読み出し、その読み出しが完了した時点で、アクセス制御信号1011を用いて制御部107にプリフェッチ動作の終了を通知する。

【0014】従って、プロセッサユニット101からプリフェッチ要求があった場合には、プロセッサユニット101の要求するラインデータのラインフィル動作と、次のラインデータのプリフェッチ制御とが可能である。

【0015】図14に従来のデータプリフェッチ制御回路のタイムチャートを示す。

【0016】

【発明が解決しようとする課題】しかしながら、上記のプリフェッチ制御回路では、プロセッサからプリフェッチ要求があった場合に、キャッシュメモリに取込まれるデータは、ラインフィル動作に基いてフェッチされたプロセッサの要求するラインデータと、プリフェッチ制御に基いてフェッチされた次のラインデータとの2つのラインデータであるため、この2つのラインデータの処理をプロセッサが終了した時点で再度プリフェッチ要求を行う必要があって、プリフェッチ要求を頻繁に繰返さなければならない欠点があった。

【0017】更に、プロセッサがフェッチするデータの性質が処理データであった場合に、プリフェッチの対象となる変数データ列は、1変数データ列であるよりも、

多変数データ列である可能性のほうが高いものの、上記プリフェッチ制御回路では、1回のプリフェッチ要求に対して1変数データ列のプリフェッチ動作しかできないため、多変数データ列のプリフェッチ動作を行わせるには、そのデータ列の変数の数だけ繰返しプリフェッチ要求する必要が生じると共に、1変数データ列に対するラインフィル動作に続いてプリフェッチ動作が実行されている際に、他の変数データ列に対するラインフィル動作が要求されても、そのラインフィル動作は直ちに開始されず、プロセッサユニット101のストール状態が長く継続し、計算機システム全体の性能の向上にも限度があった。

【0018】次に、上述の2つの欠点を具体的に説明する。例えば、流体力学や電気回路等のシミュレーションを計算機で行う場合に、次に示すサブルーチン（以下の説明では、C言語を用いる）を実行するときを例に挙げて説明する。

【0019】

```
sub0 {
    int i;
    for(i=0; i <= N; ++i) {
        func(x[i], y[i]);
    }
    return;
}
```

上記サブルーチンでは、関数funcの処理を行う必要上、1ループ毎に配列変数x,yのデータロードを実行する必要がある。キャッシュユニット102に配列変数x,yが全く存在しない場合に、 $i=0$ の時、プロセッサユニット101は、 $x[0]$ のデータロードと共に $x[0+\alpha]$ のプリフェッチ要求も行うが、 $x[0]$ のデータロードはキャッシュミスとなるため、プロセッサユニット101はストール状態になる。その後、ラインフィル動作に基いた $x[0]$ のデータロードが終了した時点で、プロセッサユニット101はストール状態から復帰して、次の処理を行うと共に、キャッシュユニット102では、 $x[0+\alpha]$ のデータプリフェッチ動作が実行される。また、 $y[0]$ のデータロードと $y[0+\alpha]$ のデータプリフェッチ動作に関しても上記 $x[0]$ の場合と同様である。

【0020】しかしながら、 $x[0]$ のデータロードによるストール状態からの復帰直後に $y[0]$ のデータロード要求が行われた場合には、上記 $x[0]$ のデータロードに続いて $x[0+\alpha]$ のプリフェッチ動作が実行されている関係上、このプリフェッチ動作が終了するまで、 $y[0]$ のデータロードを実行することができず、このため、 $y[0]$ のストール時間は、上記 $x[0+\alpha]$ のデータプリフェッチ時間と、 $y[0]$ のデータロード時間とを合わせた時間となり、プロセッサユニット101を効率的に動作させることができない。

【0021】また、 $1 \leq i \leq (2\alpha-1)$ では、キャッ

シュユニット102には配列変数x,yが存在するので、キャッシュヒットになり、プロセッサユニット101はストール状態にはならないが、 $i=2\alpha$ のループでは、キャッシュユニット102には配列変数x,yが存在しないため、 $i=0$ の場合と同様に、プロセッサ101はストール状態に陥ることになる。即ち、 $i$ が $2\alpha$ 増加する毎に、プリフェッチ動作の再要求を行う必要が生じると共に、配列変数x,yのデータロードによるプロセッサユニット101のストール状態が長く継続する欠点がある。

【0022】尚、上記のサブルーチン処理を上述の従来のデータプリフェッチ制御回路により動作させた場合のタイムチャートを図15に示す。

【0023】本発明は上記問題点を鑑みてなされたものであり、その目的は、データプリフェッチ制御回路において、1回のプリフェッチ要求で多数のラインデータのプリフェッチを可能とすると共に、多変数データ列のプリフェッチ動作を可能とすることにある。

【0024】

【課題を解決するための手段】上記問題点を解決するため、本発明では、プロセッサから発行されるデータプリフェッチアドレスをキューイングすることにより、多変数データ列のプリフェッチを実現すると共に、そのキューイングしたプリフェッチアドレスを繰返し自動更新できる構成とすることにより、1回のプリフェッチ要求で多数のラインデータのプリフェッチを可能にする。

【0025】すなわち、請求項1記載の発明のデータプリフェッチ制御回路は、プロセッサユニットが使用するデータをキャッシュユニットにプリフェッチするデータプリフェッチ制御回路であって、プリフェッチアドレスを格納するプリフェッチアドレスキューと、上記キャッシュユニットのラインサイズ値を格納するラインサイズ値格納手段と、上記プリフェッチアドレスキューに格納されたプリフェッチアドレス及び上記ラインサイズ値格納手段に格納されたラインサイズ値に基いて、プリフェッチアドレスを更新する演算手段と、上記プロセッサユニットから出力されるプリフェッチアドレス及び上記演算手段により更新されたプリフェッチアドレスのうち何れか一方を選択する第1及び第2の選択手段とを備え、上記第1の選択手段は、選択した値を上記プリフェッチアドレスキューに出力し、上記第2の選択手段は、選択した値を上記キャッシュユニットに出力し、更に、上記プロセッサユニットからプリフェッチアドレスが出力されるときこのプリフェッチアドレスを選択し、その後は、更新されたプリフェッチアドレスを選択するように、上記第1及び第2の各選択手段を制御する制御手段とを備えている構成である。

【0026】また、請求項2記載の発明では、上記請求項1記載のデータプリフェッチ制御回路において、ラインサイズ値格納手段は、予め、所定の1種類のラインサ

イズ値が記憶されたレジスタにより構成される構成である。

【0027】更に、請求項3記載の発明では、上記請求項1記載のデータプリフェッチ制御回路において、ラインサイズ値格納手段は、ラインサイズ値が外部から任意に与えられて、ラインサイズ値が可変である構成である。

【0028】加えて、請求項4記載の発明では、上記請求項3記載のデータプリフェッチ制御回路において、ラインサイズ値格納手段は、プリフェッチアドレスの増分値を格納する増分値格納キューと、プロセッサユニットから出力されるオフセットデータ及び上記増分値格納キューに格納したプリフェッチアドレスの増分値のうち何れか一方を選択し、その選択した値を上記増分値格納キューに出力する第3の選択手段と、プロセッサユニットからオフセットデータが出力されるときそのオフセットデータを選択し、その後は、増分値格納キューに格納したプリフェッチアドレスの増分値を選択するよう、上記第3の選択手段を制御する第2の制御手段とから成る構成である。

【0029】更に加えて、請求項5記載の発明では、上記請求項1記載のデータプリフェッチ制御回路において、プリフェッチアドレスキューは、FIFOメモリである構成である。

【0030】また、請求項6記載の発明では、上記請求項1、請求項2及び請求項4記載のデータプリフェッチ制御回路において、プロセッサユニットは、プリフェッチ中断信号を出力し、制御手段は、上記プロセッサユニットからのプリフェッチ中断信号を受けて、プリフェッチアドレスキューに格納されているプリフェッチアドレスを消去する構成である。

【0031】更に、請求項7記載の発明では、上記請求項1、請求項2及び請求項4記載のデータプリフェッチ制御回路において、プリフェッチ回数をカウントし、プリフェッチ回数が設定回数に達したとき、プリフェッチ中断信号を出力する中断信号出力手段を備え、制御手段は、上記中断信号出力手段から出力されるプリフェッチ中断信号を受けて、プリフェッチアドレスキューに格納されているプリフェッチアドレスを消去する構成である。

【0032】加えて、請求項8記載の発明では、上記請求項7記載のデータプリフェッチ制御回路において、中断信号出力手段は、プリフェッチの実行残数を格納する実行残数格納手段と、上記実行残数格納手段から出力される値をゼロ比較する比較手段と、上記実行残数格納手段に格納されたプリフェッチの実行残数から1を減算する減算手段と、プロセッサユニットから出力されるプリフェッチ回数及び上記減算手段により減算されたプリフェッチの実行残数のうち何れか一方を選択し、その選択した値を上記実行残数格納手段に出力する第4の選択手

段と、上記プロセッサユニットからプリフェッチ回数が出力されるときそのプリフェッチ回数を選択し、その後は、減算手段により減算されたプリフェッチの実行残数を選択するよう、上記第4の選択手段を制御する第3の制御手段とから成る構成を採用している。

【0033】

【作用】以上の構成により、請求項1、請求項2及び請求項5記載の発明では、プロセッサユニットからプリフェッチアドレスが出力された場合には、このアドレスがプリフェッチアドレスキューに格納され、この格納されたプリフェッチアドレスに基づいてキャッシュユニットがラインフィル動作及びプリフェッチ動作を行うので、多変数データ列のラインフィル動作及びプリフェッチ動作が可能である。

【0034】しかも、上記格納されたプリフェッチアドレスは、演算手段によりラインサイズ値格納手段のラインサイズ分だけ何回も更新され、この更新されたプリフェッチアドレスに基づいて制御手段がキャッシュユニットを制御することを繰返すので、プロセッサユニットからの1回のプリフェッチ要求に対して、何回でもデータのプリフェッチが可能となる。

【0035】また、請求項3及び請求項4記載の発明では、ラインサイズ値格納手段に格納されるラインサイズ値が、外部から任意に与えられて、可変であるので、変数データ列毎に規則的に離散化された変数データ列や、プロセッサユニットのアクセスする変数データ列のアドレスがマイナス方向に増加する場合であっても、そのプリフェッチ動作が可能になる。

【0036】更に、請求項6、請求項7及び請求項8記載の発明では、プロセッサユニット又は中断信号出力手段からプリフェッチ中断信号が出力されるので、不必要なプリフェッチ動作を回避することができる。

【0037】

【実施例】以下、本発明の実施例のデータプリフェッチ制御回路について、図面を参照しながら、説明する。

【0038】(第1の実施例) 図1は本発明の第1の実施例におけるデータプリフェッチ制御回路の概略構成図を示す。

【0039】同図において、11はプロセッサユニット、12はキャッシュユニット、13はキャッシュのラインサイズ値を予め格納しているラインサイズ格納手段を構成するラインサイズレジスタである。

【0040】また、14はプリフェッチアドレスを格納するFIFO (first in firstout)メモリであるプリフェッチアドレスキュー、15は上記プリフェッチアドレスキュー14の出力及びラインサイズレジスタ13の出力から新しい(更新された)プリフェッチアドレスの計算を行なう演算手段としての加算器である。

【0041】更に、17は、プロセッサユニット11から出力されるアドレスと、加算器15の出力とのうち何

れか一方を選択し、その選択した出力をプリフェッチアドレスキュー14に出力する第1の選択手段としてのマルチプレクサ、16は、上記プロセッサユニット11から出力されるアドレスと、加算器15の出力とのうち何れか一方を選択し、その選択した出力をキャッシュユニット12に出力する第2の選択手段としてのマルチプレクサである。

【0042】加えて、18はプリフェッチ動作を制御する制御手段としての制御部、19はプリフェッチアドレスキュー14を制御する制御信号、110はマルチプレクサ17の選択信号、111はマルチプレクサ16の選択信号、112はキャッシュユニット12と制御部18とでハンドシェークされるアクセス制御信号、113はプロセッサユニット11と制御部18とでハンドシェークされるアクセス制御信号、114はプロセッサユニット11から制御部18に出力されるプリフェッチ要求信号、115はプロセッサユニット11から制御部18に出力されるプリフェッチ終了信号、116はプロセッサユニット11から制御部18に出力されるプリフェッチ一時中断信号、117はプロセッサユニット11から制御部18に出力されるプリフェッチ再開信号である。

【0043】更に加えて、118はプロセッサユニット11から出力されるデータバス、119はプロセッサユニット11から出力されるアドレスバスである。

【0044】次に、上記制御部18の構成を説明する。

【0045】プロセッサユニット11からプリフェッチ要求信号114を用いてデータのプリフェッチ要求がなされた場合には、制御部18は、マルチプレクサ16、17がプロセッサユニット11から出力されるアドレスバス119を選択するように選択信号110、111を設定することにより、キャッシュユニット12及びプリフェッチアドレスキュー14に対して、アドレスバス119上のプリフェッチアドレスを通知する。

【0046】次に、制御部18は、キャッシュユニット12に対してプリフェッチアドレスに対するラインフィル動作をアクセス制御信号112を用いて要求すると共に、プリフェッチアドレスキュー14に対しては、プリフェッチアドレスのキューイングを制御信号19を用いて実行する。

【0047】更に、制御部18は、プロセッサユニット11からのアクセス要求及びキャッシュユニット12のラインフィル動作から、主メモリのバスが共に開放されている場合には、プリフェッチアドレスキュー14内のプリフェッチアドレスデータの有無を制御信号19を用いて調べ、プリフェッチアドレスデータが存在するときには、プリフェッチアドレスキュー14からプリフェッチアドレスデータを取り出して加算器15に出力させ、上記加算器15で前記取出されたプリフェッチアドレスデータとラインサイズレジスタ13の出力値とを合計して更新した新たなプリフェッチアドレスの計算を行なわ

せる。更に、マルチプレクサ16、17が加算器15の出力を選択するように選択信号110、111を設定し、キャッシュユニット12に対して上記更新されたプリフェッチアドレスに対するプリフェッチ動作をアクセス制御信号112を用いて要求すると共に、プリフェッチアドレスキュー14に対しては、上記更新されたプリフェッチアドレスの再度のキューイングを制御信号19を用いて実行する。

【0048】また、制御部18は、プリフェッチ一時中断信号116及びプリフェッチ再開信号117を用いて、キャッシュシステム12にプリフェッチされるデータ量とプロセッサユニット11におけるデータ処理量との関係がほぼ均衡するように制御すると共に、プロセッサユニット11からのプリフェッチ中断信号115によりデータのプリフェッチ中断の要求がなされた場合には、制御信号19を用いて、プリフェッチアドレスキュー14内のプリフェッチアドレスを消去するように制御する。

【0049】本実施例のデータプリフェッチ制御回路において、次のサブルーチンを実行した場合のタイムチャートを図2及び図3に、プリフェッチアドレスキュー14の動作を図4に各々示す。

【0050】

```
sub0 {
    int i;
    for(i=0; i (<= N; ++i) {
        func(x[i], y[i]);
    }
    return;
}
```

したがって、本実施例においては、プリフェッチアドレスキュー14と、加算器15とにより、プリフェッチアドレスを更新し、この更新したプリフェッチアドレスを再度プリフェッチアドレスキュー14にキューイングすることがプリフェッチ中断要求があるまで繰返されるので、ラインフィル動作の終了後は、1回のプリフェッチ要求に対して何回でもプリフェッチ動作を繰返すことができ、プリフェッチ要求を頻繁に繰返す必要がない。

【0051】しかも、プリフェッチアドレスキュー14へのプリフェッチアドレスのキューイングにより、多変数データ列のプリフェッチ動作を実現することができる。

【0052】(第2の実施例)以下、本発明の第2の実施例について図面を参照しながら説明する。

【0053】図5は本発明の第2の実施例を示すデータプリフェッチ制御回路の概略構成図を示す。

【0054】同図において、41はプロセッサユニット、42はキャッシュユニット、43はプロセッサユニット41から出力されるデータバス、44はプリフェッチアドレスを格納するプリフェッチアドレスキュー、4

5はプリフェッチアドレスの増分値を格納する増分値格納キューとしてのオフセットキュー、46はプリフェッチアドレスキュー44とオフセットキュー45の出力から新しいプリフェッチアドレスの計算を行なう加算器である。

【0055】また、47はプロセッサユニット41から出力されるアドレスと、加算器46の出力とのうち何れか一方を選択し、その選択した出力をキャッシュユニット42に出力するマルチプレクサ、48は、データバス43とオフセットキュー45の出力とのうち何れか一方を選択し、その選択した出力をオフセットキュー45に出力する第3の選択手段としてのマルチプレクサ、49はプロセッサユニット41から出力されるアドレスと加算器46の出力とのうち何れか一方を選択し、その選択した出力をプリフェッチアドレスキュー44に出力するマルチプレクサである。

【0056】更に、410はプリフェッチ動作を制御する制御部であって、本実施例では請求項2記載の第2の制御手段としても機能する。411はマルチプレクサ49への選択信号、412はマルチプレクサ48への選択信号、413はプリフェッチアドレスキュー44を制御する制御信号、414はオフセットキュー45を制御する制御信号、415はマルチプレクサ47への選択信号、416はプロセッサユニット41と制御部410とでハンドシェークされるアクセス制御信号、417はキャッシュユニット42と制御部410とでハンドシェークされるアクセス制御信号である。また、418はプロセッサユニット41から制御部410に出力されるプリフェッチ要求信号、419はプロセッサユニット41から制御部410に出力されるプリフェッチ終了信号、420はプロセッサユニット41から制御部410に出力されるプリフェッチ一時中断信号、421は、プロセッサユニット41から制御部410に出力されるプリフェッチ再開信号、422はプロセッサユニット41から出力されるアドレスバスである。

【0057】次に、上記制御部410による制御を説明する。

【0058】プロセッサユニット41からプリフェッチ要求信号418を用いてデータのプリフェッチ要求がなされた場合には、制御部410は、マルチプレクサ47、49がプロセッサユニット41からのアドレスバス422を選択するように選択信号411、415を設定すると共に、マルチプレクサ48がデータバス43を選択するように選択信号412を設定することにより、キャッシュユニット42及びプリフェッチアドレスキュー44に対して、アドレスバス422上のプリフェッチアドレスを通知し、同時に、プロセッサユニット41からデータバス43に出力されるオフセットデータをオフセットキュー45に通知する。

【0059】次に、制御部410は、キャッシュユニッ

ト42に対しては、アクセス制御信号417を用いて、プリフェッチアドレスに対するラインフィル動作を要求し、プリフェッチアドレスキュー44に対しては、制御信号413を用いて、プリフェッチアドレスのキューイングを実行し、オフセットキュー45に対しては、制御信号414を用いて、オフセットデータのキューイングを実行する。

【0060】更に、制御部410は、プロセッサユニット41からのアクセス要求、及びキャッシュユニット42のラインフィル動作から、主メモリのバスが共に開放されている場合には、プリフェッチアドレスキュー44のプリフェッチアドレスデータの有無を制御信号413を用いて調べ、プリフェッチアドレスデータが存在する場合には、プリフェッチアドレスキュー44からプリフェッチアドレスデータを取り出して加算器46に出力させると共に、オフセットキュー45からオフセットデータを取り出して加算器46に出力させ、これにより、上記加算器46において、更新された新たなプリフェッチアドレスの計算を行なわせる。また、マルチプレクサ47、49が加算器46の出力を選択するように選択信号411、415を設定すると共に、マルチプレクサ48がオフセットキュー45の出力を選択するように選択信号412を設定することにより、キャッシュユニット42に対しては、アクセス制御信号417を用いて、更新されたプリフェッチアドレスに対するプリフェッチ動作を要求し、プリフェッチアドレスキュー44に対しては、制御信号413を用いて、更新されたプリフェッチアドレスの再度のキューイングを実行し、オフセットキュー45に対しては、制御信号412を用いて、オフセットデータの再度のキューイングを実行する。

【0061】更に、制御部410は、プリフェッチ一時中断信号420及びプリフェッチ再開信号421を用いて、キャッシュシステム42にプリフェッチされるデータ量と、プロセッサユニット41におけるデータ処理量との関係が均衡するように制御すると共に、プロセッサユニット41がプリフェッチ終了信号419を用いてデータのプリフェッチ中断要求をしている場合には、制御信号413、414を用いて、プリフェッチアドレスキュー44内のプリフェッチアドレス及びオフセットキュー45内のオフセットデータを消去する。

【0062】上記オフセットキュー45、マルチプレクサ48及び制御部410により、プリフェッチされるデータのラインサイズが外部から任意に与えられて、可変となるラインサイズ値格納手段を構成する。

【0063】本実施例のデータプリフェッチ制御回路において、次のサブルーチンを実行した場合のタイムチャートを図6及び図7に、プリフェッチアドレスキュー44及びオフセットキュー45の動作を図8に各々示す。

【0064】

13

```

sub0 {
    int i;
    for(i=0; i <= N; ++i) {
        func(x[i], y[N-i]);
    }
    return;
}

```

以上のように、本実施例では、プリフェッチアドレスキュー44と、プリフェッチアドレスの増分値を格納するオフセットキュー45と、加算器46とを用いて、プリフェッチアドレスを更新し、この更新されたプリフェッチアドレスを再度プリフェッチアドレスキュー44にキューイングすると共に、更新されたプリフェッチアドレスに対応するオフセットデータを再度キューイングしたので、多変数データ列のプリフェッチ動作と、変数データ列毎に規則的に離散化された変数データ列や、プロセッサユニット41のアクセスする変数データ列のアドレスがマイナス方向に増加する場合でのプリフェッチ動作とを可能にすることができる。

【0065】(第3の実施例)以下、本発明の第3の実施例について図面を参照しながら説明する。

【0066】図9は本発明の第3の実施例を示すデータプリフェッチ制御回路の概略構成図である。

【0067】同図において、71はプロセッサユニット、72はキャッシュユニット、73はプロセッサユニット71から出力されるデータバス、74はプリフェッチアドレスを格納するプリフェッチアドレスキュー、75はプリフェッチアドレスの増分値を格納するオフセットキュー、76はプリフェッチアドレスキュー74とオフセットキュー75の出力とに基づいて新しいプリフェッチアドレスの計算を行なう加算器である。

【0068】また、77は、プロセッサユニット71から出力されるアドレスと、加算器76の出力とのうち何れか一方を選択し、その選択した出力をキャッシュユニット72に出力するマルチプレクサ、78はデータバス73とオフセットキュー75の出力とのうち何れか一方を選択し、その選択した出力をオフセットキュー75に出力するマルチプレクサ、79はプロセッサユニット71から出力されるアドレスと加算器76の出力とのうち何れか一方を選択し、その選択した出力をプリフェッチアドレスキュー74に出力するマルチプレクサである。

【0069】更に、710はプリフェッチ回数を格納する実行残数格納手段としてのプリフェッチカウンタキュー、711はプリフェッチカウンタキュー710の出力がゼロか否かを比較し、判定する比較手段としてのゼロ比較器、712はプリフェッチカウンタキュー710の出力を-1だけ減算する減算手段としての-1減算器、713はデータバス73と-1減算器712の出力とのうち何れか一方を選択し、その選択した出力をプリフェッチカウンタキュー710に出力する第4の選択手段と

14

してのマルチプレクサである。

【0070】加えて、714はプリフェッチ動作を制御する制御部であって、本実施例では請求項3記載の第3の制御手段としても機能する。715はマルチプレクサ79への選択信号、716はマルチプレクサ78への選択信号、717はマルチプレクサ713への選択信号、718はプリフェッチアドレスキュー74を制御する制御信号、719はオフセットキュー75を制御する制御信号、720はプリフェッチカウンタキュー710を制御する制御信号、721はマルチプレクサ77への選択信号、722は制御部714に入力されるゼロ比較器711のゼロ判定信号である。

【0071】更に、723はプロセッサユニット71と制御部714とでハンドシェークされるアクセス制御信号、724はキャッシュユニット72と制御部714とでハンドシェークされるアクセス制御信号、725はプロセッサユニット71から制御部714に出力されるプリフェッチ要求信号、726はプロセッサユニット71から制御部714に出力されるプリフェッチ終了信号、727は、プロセッサユニット71から制御部714に出力されるプリフェッチ一時中断信号、728はプロセッサユニット71から制御部714に出力されるプリフェッチ再開信号、729はプロセッサユニット71から出力されるアドレスバスである。

【0072】上記制御部714による制御を説明する。

【0073】プロセッサユニット71がプリフェッチ要求信号725を用いてデータのプリフェッチ要求をしている場合には、制御部714は、マルチプレクサ77、79がプロセッサユニット71から出力されるアドレスバス729を選択するように選択信号715、721を設定すると共に、マルチプレクサ78、713がデータバス73を選択するように選択信号716、717を設定することにより、キャッシュユニット72とプリフェッチアドレスキュー74に対して、アドレスバス729上のプリフェッチアドレスを通知し、同時に、プロセッサユニット71からデータバス73に出力されるオフセットデータをオフセットキュー75に通知し、更にプロセッサユニット71からデータバス73に出力されるプリフェッチカウンタ値をプリフェッチカウンタキュー710に通知する。

【0074】更に、制御部714は、キャッシュユニット72に対しては、アクセス制御信号724を用いて、プリフェッチアドレスに対するラインフィル動作を要求し、プリフェッチアドレスキュー74に対しては、制御信号718を用いて、プリフェッチアドレスのキューイングを実行し、オフセットキュー75に対しては、制御信号719を用いてオフセットデータのキューイングを実行し、プリフェッチカウンタキュー710に対しては、制御信号720を用いて、プリフェッチカウンタ値のキューイングを実行する。



15

【0075】加えて、制御部714は、プロセッサユニット71からのアクセス要求、及びキャッシュユニット72のラインフィル動作から、主メモリのバスが共に開放されている場合には、プリフェッチアドレスキュー74内のプリフェッチアドレスデータの有無を制御信号718を用いて調べ、プリフェッチアドレスデータが存在するときには、プリフェッチアドレスキュー74からプリフェッチアドレスデータを取り出して加算器76に出力させると共に、オフセットキュー75からオフセットデータを取り出して加算器76に出力させることにより、上記加算器76において、新たなプリフェッチアドレスの計算を行なう。

【0076】更に、制御部714は、プリフェッチカウントキュー710からプリフェッチ残数データを取り出して、ゼロ比較器311にてこの残数データのゼロ判定を行なう。この際、ゼロ比較器711にてゼロ判定がなされない場合には、マルチプレクサ77、79が加算器76の出力を選択するように選択信号715、721を設定すると共に、マルチプレクサ78がオフセットキュー75の出力を選択するように選択信号716を設定し、更にマルチプレクサ713が-1減算器712の出力を選択するように選択信号717を設定することにより、キャッシュユニット72に対しては、アクセス制御信号724を用いて、更新されたプリフェッチアドレスに対するプリフェッチ動作を要求し、プリフェッチアドレスキュー74に対しては、制御信号718を用いて、更新されたプリフェッチアドレスの再度のキューイングを実行し、オフセットキュー75に対しては、制御信号719を用いて、オフセットデータの再度のキューイングを実行し、プリフェッチカウントキュー710に対しては、制御信号720を用いて、-1減算器712にて現在のプリフェッチ残数を-1だけ減算した値の新たなプリフェッチ残数のキューイングを実行する。

【0077】一方、ゼロ比較器711にてゼロ判定がなされた場合には、そのゼロ判定がゼロ判定信号722を用いて制御部714に通知され、加算器76の更新されたプリフェッチアドレス、及びそれに対応したオフセットデータとプリフェッチ残数値が破棄される。

【0078】また、制御部714は、プリフェッチ一時中断信号727及びプリフェッチ再開信号728を用いて、キャッシュシステム72にプリフェッチされるデータ量とプロセッサユニット71におけるデータ処理量との関係が均衡するように制御すると共に、プロセッサユニット71がプリフェッチ終了信号726を用いてデータのプリフェッチ中断要求をした場合には、制御信号718、719、720を用いて、各々、プリフェッチアドレスキュー74内のプリフェッチアドレスと、オフセットキュー75内のオフセットデータと、プリフェッチカウントキュー710内のプリフェッチ残数値とを消去する。

16

【0079】上記プリフェッチカウントキュー710、ゼロ比較器711、-1減算器712、マルチプレクサ713及び制御部714により、プリフェッチ回数をカウントして、そのプリフェッチ回数が設定回数に達した時にプリフェッチ中断信号を出力する中断信号出力手段740を構成している。

【0080】本実施例のデータプリフェッチ制御回路において、次に示すサブルーチンを実行した場合のタイムチャートを図10及び図11に、プリフェッチアドレスキュー74、オフセットキュー75及びプリフェッチカウントキュー710の動作を図12に各々示す。

【0081】

```
sub0 {
    int i;
    for(i=0; i <= N; ++i) {
        func(x[i], y[N-i]);
    }
    return;
}
```

したがって、本実施例においては、特に、プリフェッチ残数を格納するプリフェッチカウントキューを設け、プリフェッチ制御を行う毎にプリフェッチ残数を-1だけ減算して上記プリフェッチカウントキューに再度キューイングするしたので、上記第2の実施例と同様に、多変数データ列のプリフェッチ動作と、変数データ列毎に規則的に離散化された変数データ列及びプロセッサユニットのアクセスする変数データ列のアドレスがマイナス方向に増加する場合でのプリフェッチ動作とを実現できると共に、更に、プリフェッチ残数をチェックして、無駄なプリフェッチ動作を無くすることができる。

【0082】

【発明の効果】以上説明したように、請求項1、請求項2及び請求項5記載の発明のデータプリフェッチ制御回路によれば、プロセッサユニットから出力されるプリフェッチアドレスをプリフェッチアドレスキューに格納すると共に、この格納したプリフェッチアドレスを繰返し自動更新したので、多変数データ列のラインフィル動作及びプリフェッチ動作を可能にできると共に、プロセッサユニットからの1回のプリフェッチ要求に対して、何回でもデータのプリフェッチが可能である。

【0083】また、請求項3及び請求項4記載の発明のデータプリフェッチ制御回路によれば、プリフェッチするデータのラインサイズ値を、外部から任意に与えて、可変としたので、変数データ列毎に規則的に離散化された変数データ列や、プロセッサユニットのアクセスする変数データ列のアドレスがマイナス方向に増加する場合であっても、そのプリフェッチ動作を可能にできる。

【0084】更に、請求項6、請求項7及び請求項8記載の発明のデータプリフェッチ制御回路によれば、プリフェッチ中断信号を出力する構成としたので、不必要な

プリフェッチ動作を回避することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施例におけるデータプリフェッチ制御回路の概略構成図である。

【図2】本発明の第1の実施例におけるデータプリフェッチ制御回路のタイムチャートを示す図である。

【図3】本発明の第1の実施例におけるデータプリフェッチ制御回路の他のタイムチャートを示す図である。

【図4】本発明の第1の実施例におけるプリフェッチアドレスキューの動作説明図である。

【図5】本発明の第2の実施例におけるデータプリフェッチ制御回路の概略構成図である。

【図6】本発明の第2の実施例におけるデータプリフェッチ制御回路のタイムチャートを示す図である。

【図7】本発明の第2の実施例におけるデータプリフェッチ制御回路の他のタイムチャートを示す図である。

【図8】本発明の第2の実施例におけるプリフェッチアドレスキューとオフセットキューの動作図である。

【図9】本発明の第3の実施例におけるデータプリフェッチ制御回路の概略構成図である。

【図10】本発明の第3の実施例におけるデータプリフェッチ制御回路のタイムチャートを示す図である。

【図11】本発明の第3の実施例におけるデータプリフェッチ制御回路の他のタイムチャートを示す図である。

【図12】本発明の第3の実施例におけるプリフェッチアドレスキューとオフセットキューとプリフェッチカウントキューの動作図である。

\*

\*【図13】従来のプリフェッチ制御回路の概略構成図である。

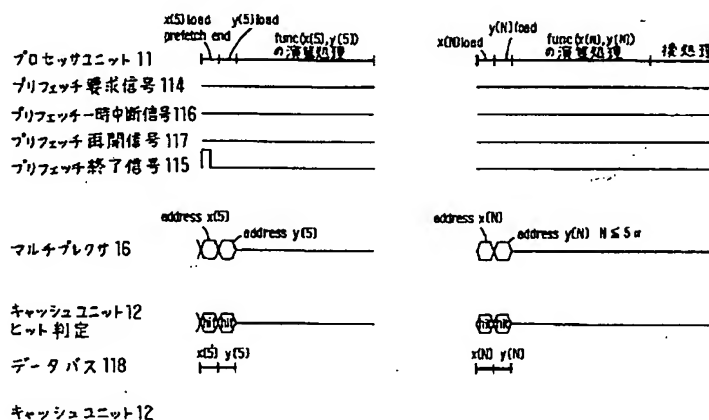
【図14】従来のプリフェッチ制御回路のタイムチャートを示す図である。

【図15】従来のプリフェッチ制御回路において関数を実行した場合のタイムチャートを示す図である。

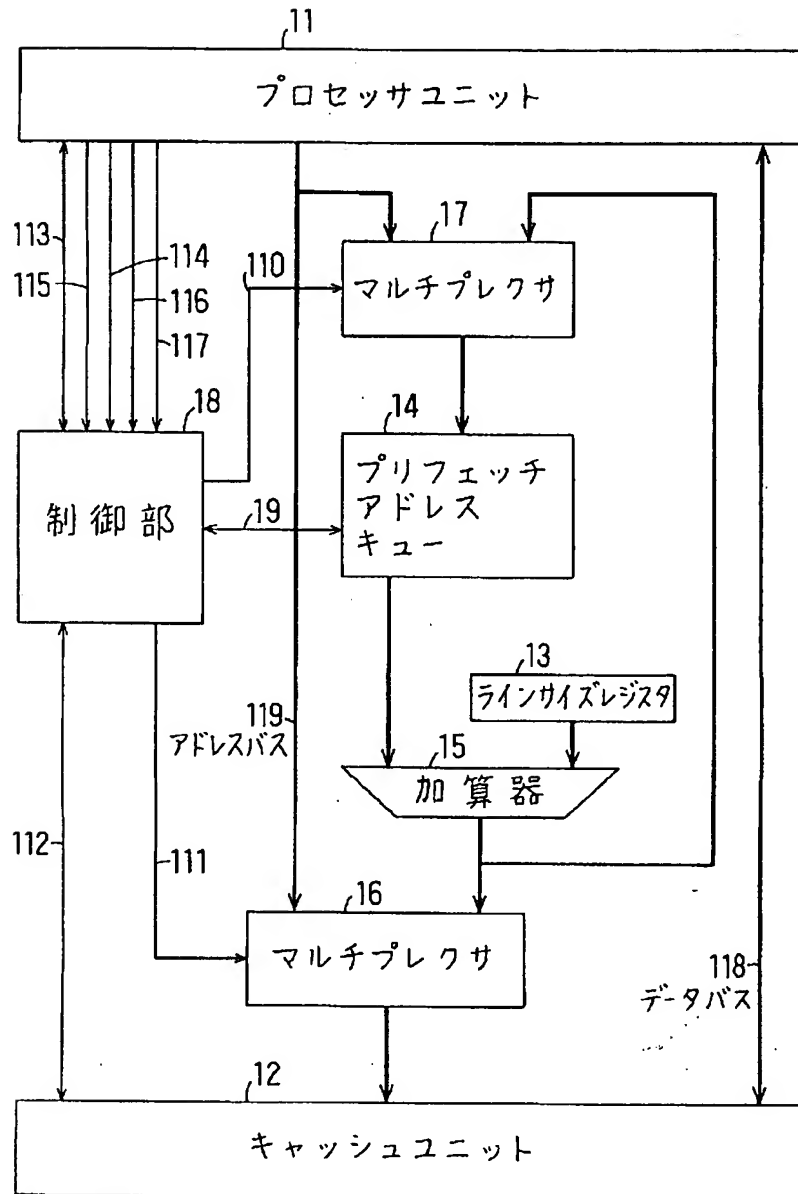
【符号の説明】

11, 41, 71,	プロセッサユニット
12, 42, 72	キャッシュユニット
10 13	ラインサイズレジスタ (ラインサイズ値格納手段)
14, 44, 74	プリフェッチアドレスキュー
15, 45, 75	加算器 (演算手段)
16	マルチプレクサ (第2の選択手段)
17	マルチプレクサ (第1の選択手段)
18	制御部 (制御手段)
45, 75	オフセットキュー (増分値格納キュー)
48	マルチプレクサ (第3の選択手段)
410	制御部 (第2の制御手段)
20 710	プリフェッチカウントキュー (実行残数格納手段)
711	ゼロ比較器 (比較手段)
712	-1 減算器 (減算手段)
713	マルチプレクサ (第4の選択手段)
714	制御部 (第3の制御手段)
740	中断信号出力手段

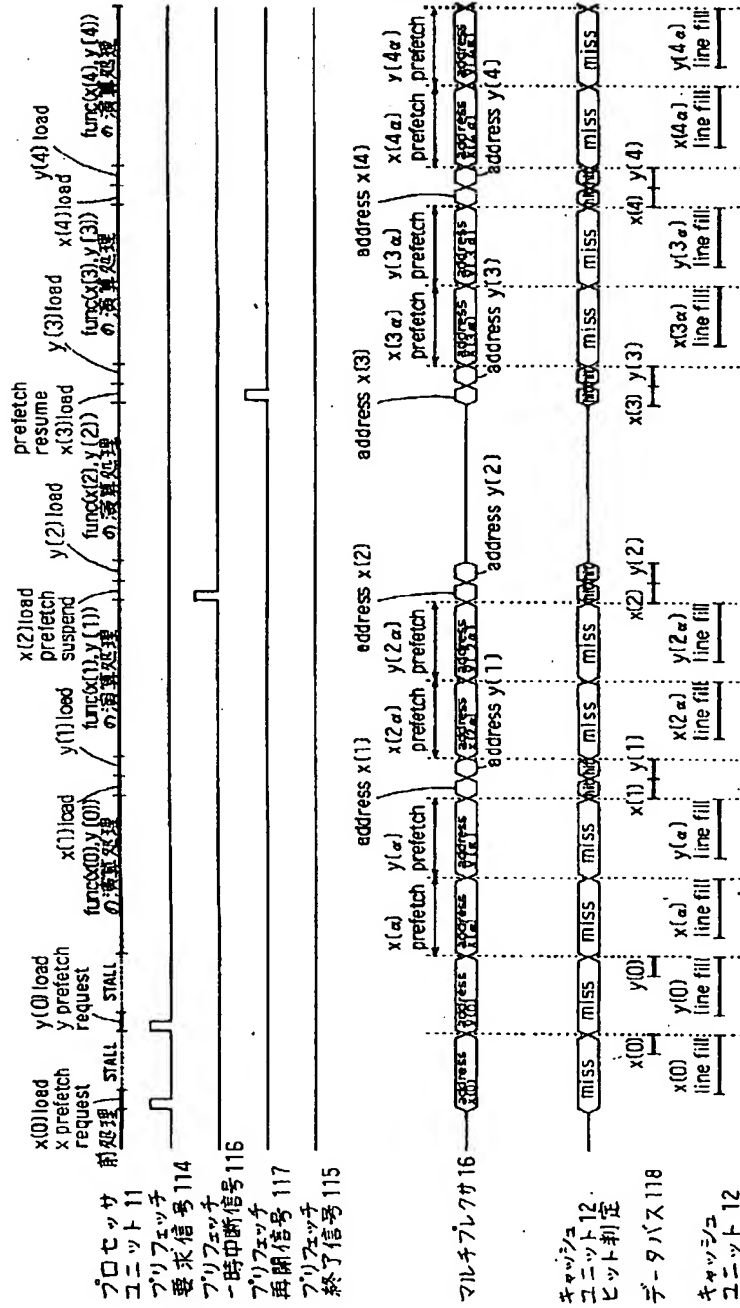
【図3】



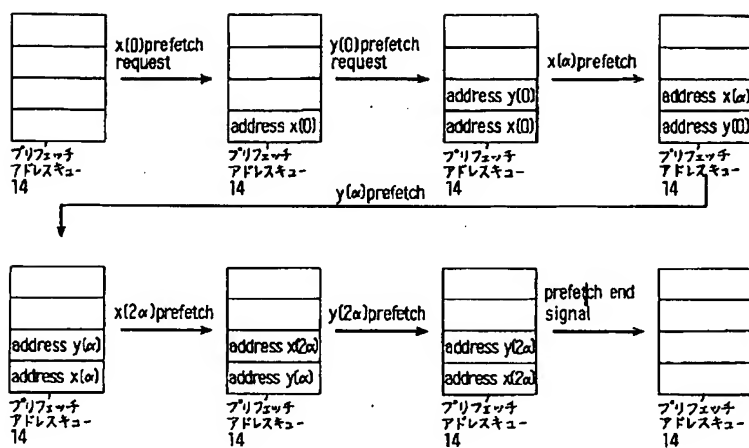
【図1】



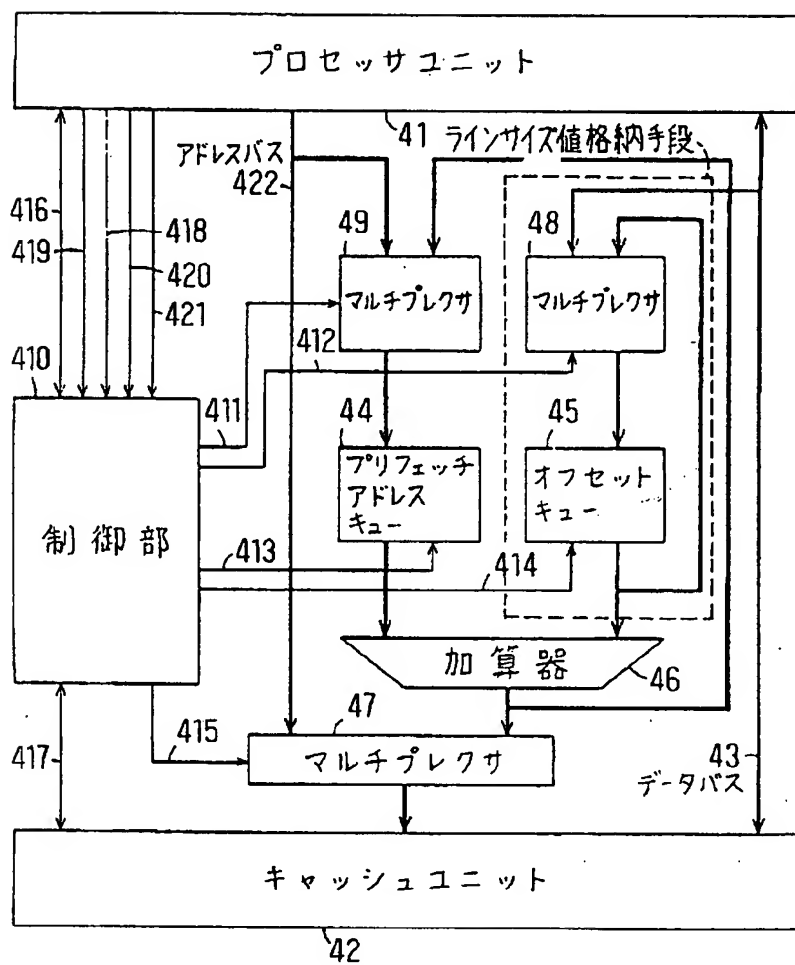
【図2】



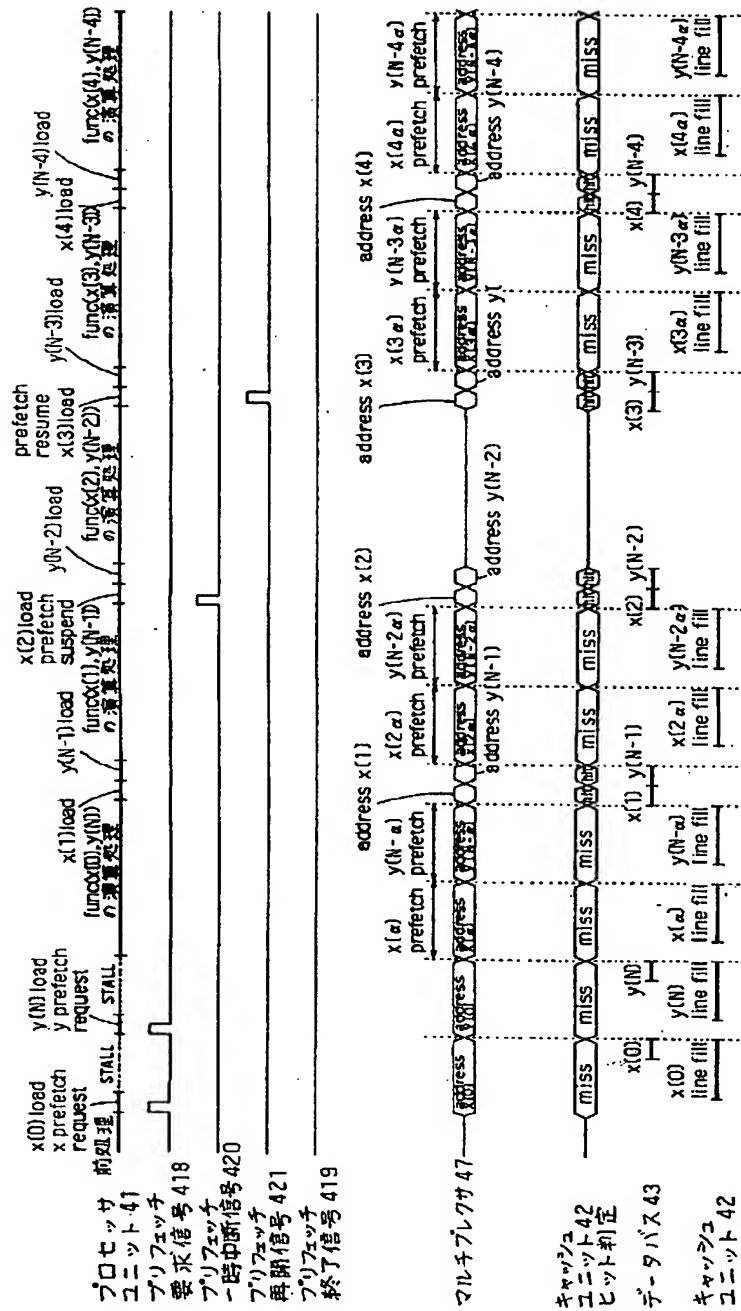
【図4】



【図5】



【図6】



**プロセッサユニット 41**

プリフェッチ要求信号 418

プリフェッチ一時中断信号 420

プリフェッチ再開信号 421

プリフェッチ終了信号 419

**マルチプレクサ 47**

**キャッシュユニット 42**

ヒット判定

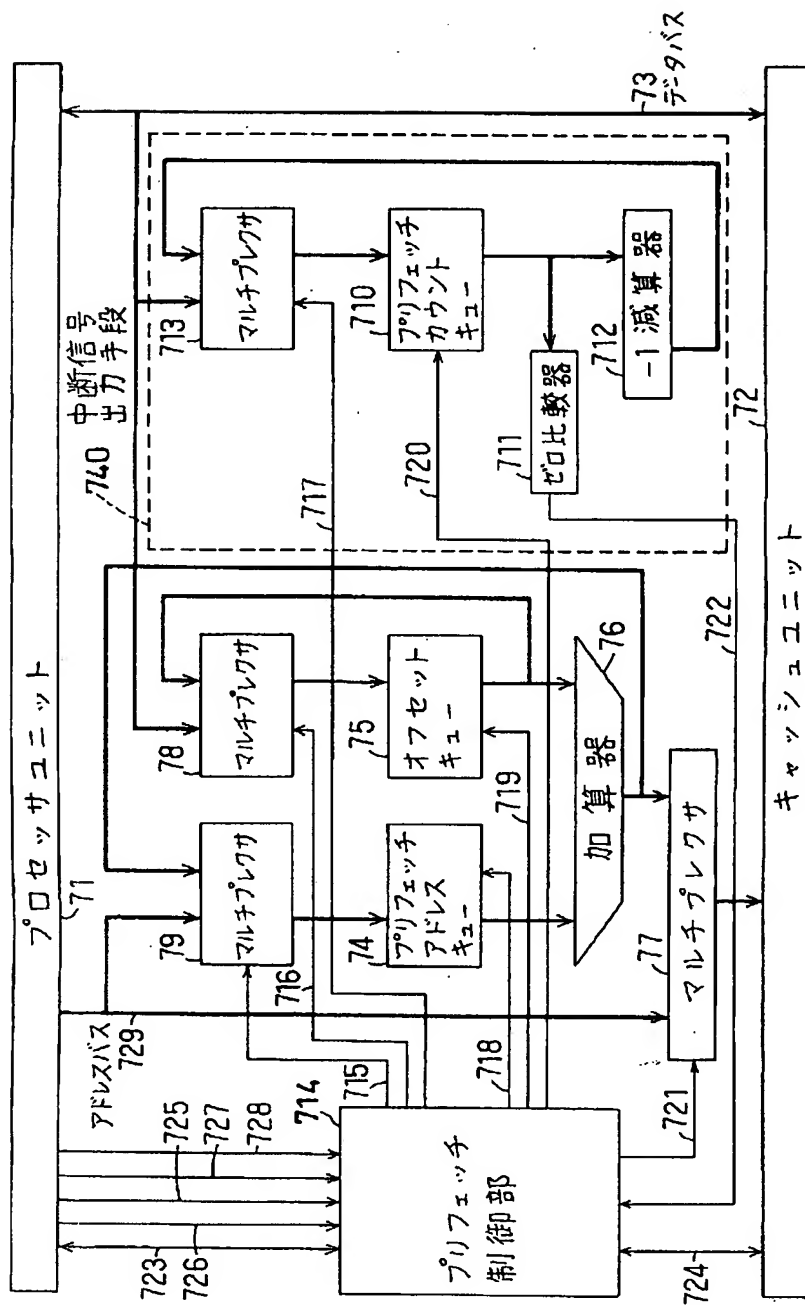
データバス 43

**キャッシュユニット 42**

Figure 1 illustrates the execution of a program on a 2x2 grid processor. The sequence of states is as follows:

- Initial State:** All cells are empty.
- Step 1:**  $x[0]$  prefetch request. The bottom-right cell contains address  $x(0)$  and value  $\alpha$ .
- Step 2:**  $y[N]$  prefetch request. The top-right cell contains address  $y(N)$  and value  $\alpha$ .
- Step 3:**  $x[\alpha]$  prefetch. The top-right cell contains address  $x(\alpha)$  and value  $\alpha$ .
- Step 4:**  $y[N-\alpha]$  prefetch. The bottom-right cell contains address  $y(N-\alpha)$  and value  $\alpha$ .
- Step 5:**  $x[2\alpha]$  prefetch. The bottom-right cell contains address  $x(2\alpha)$  and value  $\alpha$ .
- Step 6:**  $y[N-2\alpha]$  prefetch. The top-right cell contains address  $y(N-2\alpha)$  and value  $\alpha$ .
- Step 7:** prefetch end signal. All cells are empty.

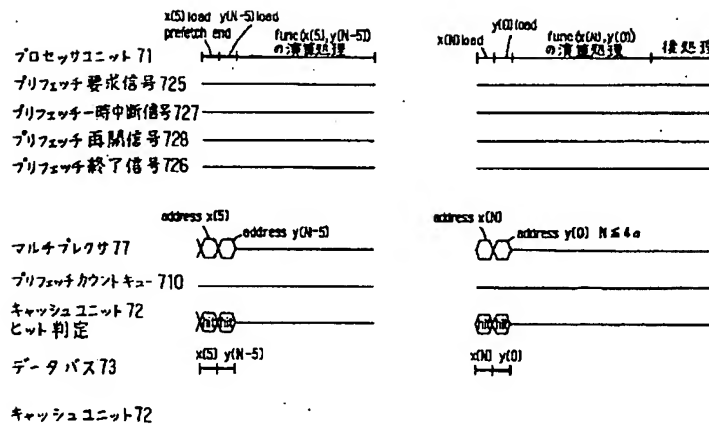
【図9】



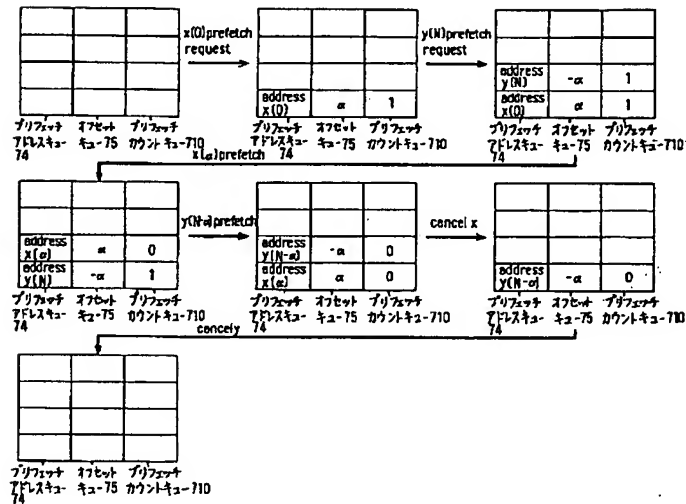


[illegible]

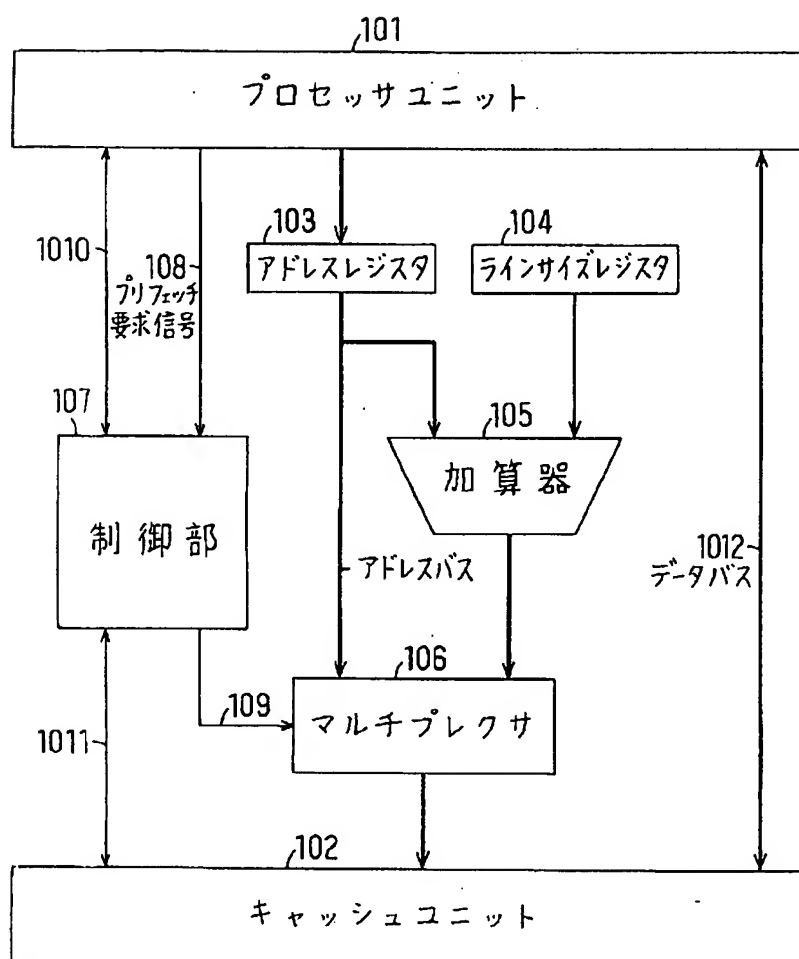
【図11】



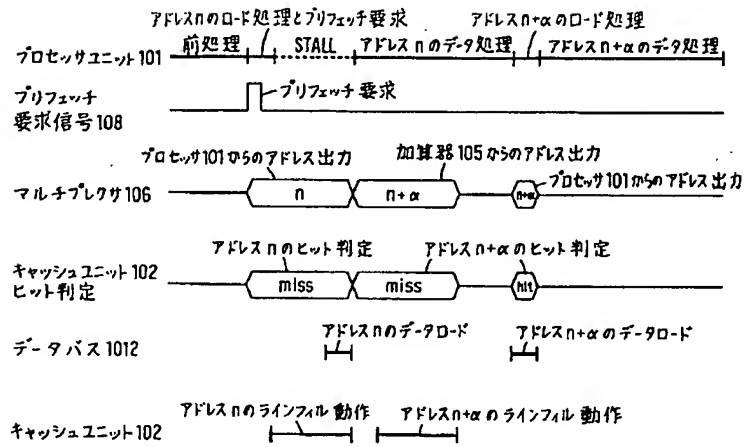
【図12】



【図13】



【図14】



$x(0)$  load  
 $x(0+\alpha)$   
 prefetch  
 request  
 前処理 STALL  
 $y(0)$  load  
 $y(0+\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha)$  load  
 $x(3\alpha)$   
 prefetch  
 request  
 STALL  
 $y(2\alpha)$  load  
 $y(3\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理  
 func  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理

プログラム  
 ユニット 101  
 プリアフェッチ  
 要求信号 108

$x(0)$  load  
 $x(0+\alpha)$   
 prefetch  
 request  
 前処理 STALL  
 $y(0)$  load  
 $y(0+\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha)$  load  
 $x(3\alpha)$   
 prefetch  
 request  
 STALL  
 $y(2\alpha)$  load  
 $y(3\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理  
 func  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理

マルチプレクサ 106  
 $x(0)$  load  
 $x(0+\alpha)$   
 prefetch  
 request  
 STALL  
 $y(0)$  load  
 $y(0+\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha)$  load  
 $x(3\alpha)$   
 prefetch  
 request  
 STALL  
 $y(2\alpha)$  load  
 $y(3\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理  
 func  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理

キャッシュ  
 ユニット 102  
 ヒット判定  
 $x(0)$  load  
 $x(0+\alpha)$   
 prefetch  
 request  
 STALL  
 $y(0)$  load  
 $y(0+\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha)$  load  
 $x(3\alpha)$   
 prefetch  
 request  
 STALL  
 $y(2\alpha)$  load  
 $y(3\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理  
 func  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理

データバス 1012  
 キャッシュ  
 ユニット 102  
 ヒット判定  
 $x(0)$  load  
 $x(0+\alpha)$   
 prefetch  
 request  
 STALL  
 $y(0)$  load  
 $y(0+\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha)$  load  
 $x(3\alpha)$   
 prefetch  
 request  
 STALL  
 $y(2\alpha)$  load  
 $y(3\alpha)$   
 prefetch  
 request  
 STALL  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理  
 func  
 $x(2\alpha+1), y(2\alpha+1)$  の演算処理